

SplashCode – A Board Game for Learning an Understanding of Algorithms in Middle School

Christiane GRESSE VON WANGENHEIM¹,
Giselle ARAÚJO E SILVA DE MEDEIROS²,
Raul MISSFELDT FILHO¹, Giani PETRI³,
Fernando DA CRUZ PINHEIRO¹, M. Nathalie F. FERREIRA⁴,
Jean C. R. HAUCK¹

¹*Department of Informatics and Statistics, Federal University of Santa Catarina (UFSC)
Florianópolis/SC, Brazil*

²*Graduate Program in Education, Federal University of Santa Catarina (UFSC)
Florianópolis/SC, Brazil*

³*Federal University of Santa Maria (UFSM)
Santa Maria/RS, Brazil*

⁴*Graduate Program in Design, Federal University of Santa Catarina (UFSC)
Florianópolis/SC, Brazil*

*e-mail: c.wangenheim@ufsc.br, giselle.medeiros@posgrad.ufsc.br,
raul.missfeldt.filho@grad.ufsc.br, gpetri@inf.ufsm.br, fernando.pinheiro@posgrad.ufsc.br,
nathalie.fortuna@posgrad.ufsc.br, jean.hauck@ufsc.br*

Received: November 2018

Abstract. Teaching computational thinking in K-12 as a 21st century skill is becoming increasingly important. Computational thinking describes a specific way of reasoning building on concepts and processes derived from algorithms and programming. One way to teach these concepts is games as an effective and efficient alternative. This article presents SplashCode, a low-cost board game to reinforce basic algorithms and programming concepts. The game was developed in a systematic way following an instructional design process, and applied and evaluated in a Brazilian public school with a total of 65 students (grade 5 to 9). First results indicate that the game can have a positive impact on motivation, learning experience, and students' learning, as well as contribute positively to social interaction, relevance, and fun. Results of this study may assist in the selection of games as an instructional strategy and/or in the development of new games for teaching computational thinking.

Keywords: game-based learning; computational thinking; algorithms and programming; K-12.

1. Introduction

In recent years, there is growing emphasis on the importance of computational thinking as a 21st century skill (Grover and Pea, 2013; Wing, 2006). Computational thinking denotes

the idea of developing a generic solution to a problem by decomposing it, identifying relevant variables and patterns, and deriving an algorithmic solution procedure (Wing, 2006). As such, computational thinking represents a cognitive ability to apply fundamental concepts and reasoning that derive from computer science in general and computer programming in particular to other domains, including real life activities (Wang, 2015). Correspondingly, computational thinking abilities specifically draw on processes such as algorithmic thinking, conditional logic, decomposition, abstraction, pattern matching, parallelization, evaluation, and generalization, thereby reflecting cognitive instantiations of concepts central to coding (Wing, 2006; Astrachan and Briggs, 2012).

Thus, fostering computational thinking early on in education seems a desirable prerequisite, preparing children for current and future demands of our knowledge societies. Diverse instructional strategies are adopted to teach computational thinking in K-12. And, although the most common strategy to teach computational thinking uses computerized activities focusing mainly on different types of programming tasks, often also unplugged activities (i.e., without the use of digital devices) are applied (Kalelioglu et al., 2016; Tsarava et al., 2018). Such activities involve the board or card games, or kinaesthetic activities that are used to represent and understand computational thinking concepts such as algorithms.

Unplugged activities, especially educational games are believed to result in a wide range of benefits, like increasing learning effectiveness, increasing interest, and motivation as well as a reduction of teaching time and instructor load (Garris et al., 2002; Gresse von Wangenheim and Shull, 2009). Games are expected to provide a fun and safe environment, where students can try alternatives and see the consequences, learning from their own mistakes and practical experiences (Pfahl et al., 2001). The unplugged approach is also an alternative to teach computational thinking in schools around the world that do not have a basic technology infrastructure (Unnikrishnan et al., 2016). Thus, educational games are supposed to be an effective and efficient instructional strategy for teaching and learning.

Besides a large number of digital games focusing on teaching computational thinking, as for example Lightbot (<https://lightbot.com/flash.html>) or Program your Robot (Kazimoglu et al., 2012), there exist also several non-digital games. These include card and board games on the educational level of K-12 aimed at learning basic concepts on algorithms and programming, besides diverse educational games used in higher education (Battistella and Gresse von Wangenheim, 2016b), as presented in Appendix A. In these games, players typically have to use action cards (representing movements, etc.) in order to command (“program”) their token across a board in order to reach a goal.

Most of these games are designed for children from 4 to 12 years. They are typically played in small groups (2–4 players) in about one hour with few very quick games of about 15 minutes. Most games are available only in English, complicating their application in other countries. We also observed that most of the games are commercial games designed as family games. Consequently, considering the cost, their application in schools may be impossible requiring for example 10 games for a class of about 40 students. Few games are directly developed for a school context. Academic publications regarding such games, presenting in a systematic way how they were developed

and evaluated, are very scarce. Thus, for most of the games, no information on how the games have been developed and/or evaluated has been encountered. The few evaluations reported ((Tsarva *et al.*, 2018) (Casarotto *et al.*, 2018) (Singh *et al.*, 2007)) were performed with participants differing from the main target audience, involving only higher education students, leaving the generalization of results to younger students (for which the games are primarily designed) questionable or at least not rigorously established (Gresse von Wangenheim and Shull, 2009; Connolly *et al.*, 2012; All *et al.*, 2016; Guzdiak, 2008).

Thus, in order provide a low-budget alternative to reinforce the understanding of basic concepts on algorithms and programming in middle school we developed a board game, SplashCode. With the purpose to obtain a systematic understanding of the quality of such games, we also systematically evaluated the game through a case study adopting the evaluation model MEEGA+Kids (Gresse von Wangenheim *et al.*, 2018), a customization of MEEGA+ (Petri *et al.*, 2018), a prominent model for evaluating the quality of games in computing education widely used in practice.

2. Research Methodology

We performed an exploratory research using a multi-method research approach as illustrated in Fig. 1.

Development of the educational game. Following EnGAGED (Battistella and Gresse von Wangenheim, 2016a), a process for developing educational games based on the in-

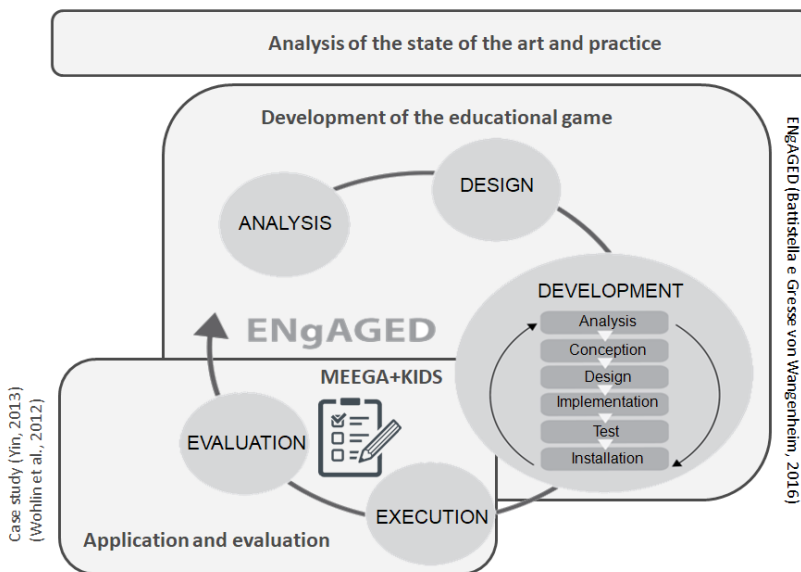


Fig. 1. Research methodology.

structional design model ADDIE (Branch, 2009), we systematically developed the game SplashCode. First we analyzed the learners and the context, and defined the instructional goal and the performance objectives. Based on this analysis, we designed the game in form of a board game. Taking into consideration several restrictions in practice, we developed the game material. During development, we conducted several test game rounds to review the game mechanics and rules and to calibrate the duration of the game.

Application and evaluation of the educational game. The game was applied within a university outreach project of the initiative Computing in School/INCoD/INE/UFSC at a public school in Florianópolis/Brazil in 2018-2. We conducted an evaluation with the purpose to analyze, if the game SplashCode allows achieving the learning objectives and if, as an instructional strategy, it provides a good player experience and usability. To achieve this goal we evaluated the game through a case study using a one-shot post-test only research design, which allows in-depth research of an individual, group or event (Wohlin *et al.*, 2012; Yin, 2017). The case study was systematically defined, planned, executed and analyzed following the process proposed by Wohlin *et al.* (2012) and Yin (2017). The case study begun with the application of the treatment (educational game SplashCode) and after the treatment data were collected. For evaluation, we use the MEEGA+KIDS model (Gresse von Wangenheim *et al.*, 2018), a model for the evaluation of games for computing education in school. MEEGA+KIDS is a customization of the MEEGA+ model (Petri *et al.*, 2018.), one of the most widely used models for the evaluation of educational games (Calderón and Ruiz, 2015). Following the MEEGA+ model, the quality of an educational game is decomposed into three quality factors (Fig. 2) and the corresponding analysis questions:

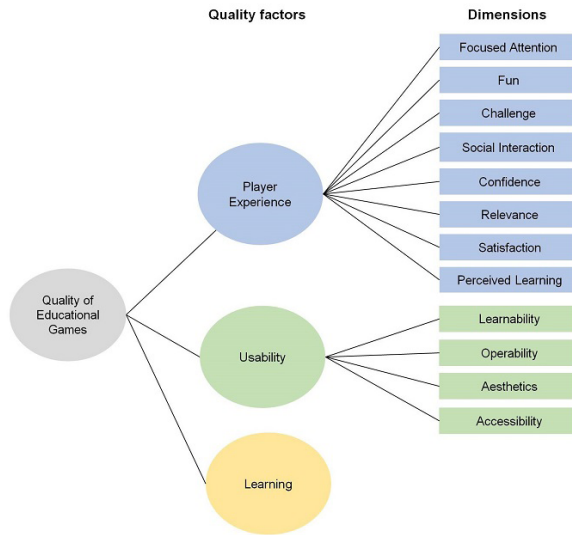
AQ1. Does the game provide a positive player experience?

AQ2. Is the game attractive and easy to play?

AQ3. Does the game contribute to the students' learning?

In this study, player experience is defined as a quality factor that covers a deep involvement of the student in the gaming task, including his/her perception of learning, feelings, pleasures, and interactions with the game, environment and other players (Savi *et al.*, 2011; O'Brien and Toms, 2010; Wiebe *et al.*, 2014; Sweetser and Wyeth, 2005; Fu *et al.*, 2009; Tullis and Albert, 2008; Keller, 1987; Sindre and Moody, 2003). Usability is defined as the degree to which a product (educational game) can be used by specified users (students) to achieve specified goals with effectiveness and efficiency in a specific context of use (computing education), being composed of the following dimensions: aesthetics, learnability, operability, and accessibility (ISO/IEC, 2014; Davis, 1989; Mohamed and Jaafar, 2010). The quality factor learning is measured in relation to the first levels of the revised version of Bloom's taxonomy (remembering, understanding and applying) (Anderson *et al.*, 2001).

Data collection is operationalized by adopting the MEEGA+KIDS self-assessment questionnaire (Gresse von Wangenheim *et al.*, 2018) answered by the students in order to collect data on their perceptions about the game on a Likert 5-point scale ranging from -2 (strongly disagree) to 2 (strongly agree). We also collected data through a brief post-test to measure learning. The collected data were analyzed in a descriptive way.



MEEGA+KIDS Questionnaire for evaluating the quality of games in K-12

Game title: *<Complete before printing>*

We would like you to answer the questions below about your perception of the quality of the game to help us improve it. All data are collected anonymously and will only be used in the context of this research. Some photos may be made as a record of this activity, but will not be published without authorization.

Researcher/Instructor's name: *<Complete before printing>*

Place and date: *<Complete before printing>*

Demographic Information	
Institution	<i><Complete before printing></i>
School year	<i><Complete before printing></i>
Course	<i><Complete before printing></i>
Age	
Gender	<input type="checkbox"/> Boy <input type="checkbox"/> Girl
How often do you play digital games?	<input type="checkbox"/> Never <input type="checkbox"/> Rarely <input type="checkbox"/> At least once a month <input type="checkbox"/> At least once a week <input type="checkbox"/> Every day
How often do you play non-digital games (card or board games, etc.)?	<input type="checkbox"/> Never <input type="checkbox"/> Rarely <input type="checkbox"/> At least once a month <input type="checkbox"/> At least once a week <input type="checkbox"/> Every day

Please, select an option according to how much you agree or disagree with each statement below.

Statements	Strongly disagree	Disagree	Neither disagree nor agree	Agree	Strongly agree
The game design is attractive (game board, cards, etc.).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The font and colors of the game match.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The size and style of fonts used in the game are easy to read.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The colors used in the game are meaningful.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Learning to play this game was easy for me.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The game rules are clear and easy to understand.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I think that the game is easy to play.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The organization of the content helped me to become confident that I would learn with this game.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
This game is appropriately challenging for me.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The game provides new challenges (offers new obstacles, situations or variations) at an appropriate pace.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The game does not become monotonous as it progresses (repetitive or boring tasks).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completing the game tasks gave me a satisfying feeling of	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 2. Decomposition of the MEEGA+KIDS model and example of the questionnaire.

3. Board Game SplashCode

Following EnGAGED (Battistella, Gresse von Wangenheim, 2016a), we systematically developed the game SplashCode as an alternative unplugged activity for teaching computational thinking.

3.1. Context Analysis

Learners: The target audience for this game is secondary school students, aged between 8 and 14 years. Typically, most students at this age already have knowledge and skills in using computers and other devices accessing the internet, social networks, digital games, videos, and music. Students also know how to use electronic devices (cell phones, computers, tablets) from home and through traditional IT literacy classes in school. They typically spend considerable time online on their cell phones, especially on social networks and/or digital games. Regarding computing-related skills, specifically programming, some students already have an understanding of computer programs, but few know how to create one. Many play video games frequently, however, few have the custom of playing non-digital games. With respect to the literature they have a preference for the manga, comics and anime books.

School environment: The teaching of computing in K-12 is being introduced in schools as part of regular classes or extracurricular activities either by focusing explicitly on teaching computing or in a multidisciplinary way integrated into other disciplines. Yet, as computing is still not part of the basic curriculum in several countries, these instructional units covering computing often are limited to a short duration. Classes typically take place in computer labs with computers (or notebooks). These computer labs are used by teachers from other knowledge areas for digital assignments teaching IT literacy, such as editing texts, making presentations, blogs, etc. Even, though, schools usually can count on an educational/IT teacher supporting the use of computers in class, most schools do not have teachers specifically trained in computing. Classes have an average of 25–40 students. Specifically, in the context of public schools there are few resources available for the acquisition of didactic material.

Curriculum: According to the K-12 Computer Science Standards (CSTA, 2017), computing education in K-12 should address computational thinking, as part of core concepts and practices (Table 1).

This includes as part of computational thinking also the concept of algorithms and programming addressing topics such as algorithms, variables, control, modularity and program development. Students should also know what an algorithm and problem solving principles are and how they work.

The game is expected to be inserted into a class aimed at teaching basic concepts of algorithms and programming as an additional instructional method to expositive lectures. This class is an initial part of teaching computational thinking covered in classes later on also through practical programming activities using block-based programming environments such as Scratch, Snap! and/or App Inventor.

Table 1
Basic computing competences in K-12 (CSTA, 2017)

Core concepts	Core practices
Computing Systems	Fostering an Inclusive Computing Culture
Networks and the Internet	Collaborating around Computing
Data and Analysis	Recognizing and Defining Computational Problems
Algorithms and Programming	Developing and Using Abstractions
Impacts of Computing	Creating Computational Artifacts
	Testing and Refining Computational Artifacts
	Communicating about Computing

3.2. Design and Development of the Game

The game is being developed to be applied in an instructional unit aimed at teaching basic concepts of computation about algorithms and programming. The purpose of the game is to reinforce the understanding of basic concepts on algorithms and programming initially presented through a lecture. We choose to adopt a game in order to provide effective and efficient learning in an engaging way. Due to practical restrictions, the maximum duration of the game is 15 minutes. The game is designed as a board game for 3–6 players from 6 years up.

The goal of the game SplashCode is to demonstrate algorithm and programming concepts as part of computational thinking. After playing the game the students will be able to:

- Decompose the steps needed to solve a problem in a precise sequence of instructions.
- Recognize that an algorithm is a set of step-by-step instructions to complete tasks.

In the context of the game’s narrative, players need to bring their pet home by moving them through a forest (the board) before the rain starts. They move their animal by means of commands like move one forward, go back, turn right, jump, etc. Winner of the game is the player who first brings the animal home.

In order to enable a wide application in public schools, a low cost game (approx. US\$2.50 for one set) has been designed in English and Brazilian Portuguese (Fig. 3). Observing the interest in mangas in this age group, Kawaii elements were chosen for the visual design of the game¹.

During game preparation, students form groups of 3–6 players. Each player chooses one of the animal tokens to be her/his character. Each player receives a programming board. At the beginning of the game, each player places their animal token in the respective field on the board. The pile of programming cards is placed in the middle of the participants. Each player receives 5 programming cards. Each player must “program” the next 3 steps on his/her programming board, placing one programming card in each space (Fig. 4).

¹ Images from freepik.com

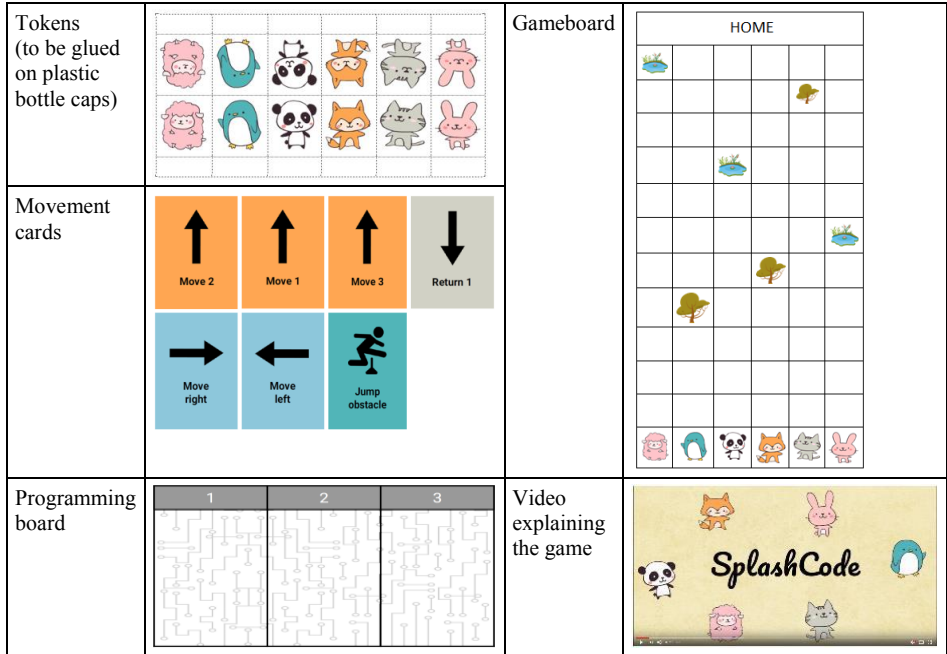


Fig. 3. SplashCode game material.



Fig. 4. Game SplashCode.

Starting with the youngest player, each player executes the next programming card from his programming board moving her/his animal token accordingly on the game board and discards the programming card. If a player collides with another animal token, tree, falls into a pool or collides with a side of the board, his/her movements are blocked, and the token remains in the same space on the game board. Turning clockwise, the other players also execute their next programming card, until all programming cards from the programming boards have been executed. Then, each player takes 3 new programming cards from the stack and programs again the next 3 programming commands on his/her programming board. These steps repeat until the game ends. The winner is the player who first gets his/her character to the “home” space of the board.

The complete game material is available in English and Brazilian Portuguese on the website of the initiative Computing in School/INCoD/INE/UFSC (<https://bit.ly/2GSZhgS>) under the Creative Commons license.

4. Application of the Game

The game SplashCode has been applied and evaluated and in the context of the outreach project Young Programming Tutors coordinated by the initiative Computing in School/INCoD/INE/UFSC with a total of 65 students at the public school Almirante Carvalhal in Florianópolis/Brazil as part of four 3-hours computing workshops in 2018 (Fig. 5).



Fig. 5. Students playing SplashCode.

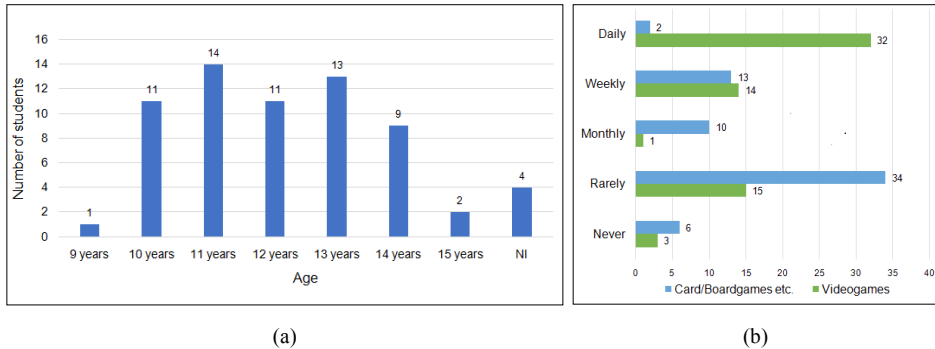


Fig. 6. (a) Age distribution of the participants. (b) Frequency of playing games.

Participants were students from middle school with exception of one 9-year-old (Fig. 6a). The majority of the students were boys (54%). Many play videogames frequently, but rarely board games (Fig. 6b.)

5. Results

The data collected in the case study were grouped into a single sample, using them cumulatively to analyze the quality of the game. Data grouping was possible due to the similarity of the research design of the applications and the standardization of data collection (Kish, 1984). In this respect, the applications were similar in terms of definition (aiming to evaluate the game in terms of player experience, usability and learning), research design (case study) and context (computing education in middle school). In addition, the measures (factors/quality dimension), method of data collection (MEEGA+KIDS questionnaire) and response format were standardized.

We present the data analysis for each of the analysis questions in the following sections.

5.1. Does the Game Provide a Positive Player Experience?

In general, the player's experience was evaluated very positive (Fig. 7).

The best evaluated dimensions are social interaction, satisfaction, fun and relevance. Students also indicated that the game gives them confidence that they are learning and that the game provides the satisfaction of moving forward through learning. The vast majority would recommend the game to their colleagues.

On the other hand, most students were not so much involved in the game that they lost the notion of time. Several also indicated that the game could be improved by creating more challenges, even being a quick game. In terms of relevance of the game to a specific course, a considerable amount of indifferent responses was given, which may be related to the fact that the game was applied in extracurricular workshops and not in the context of a specific course.

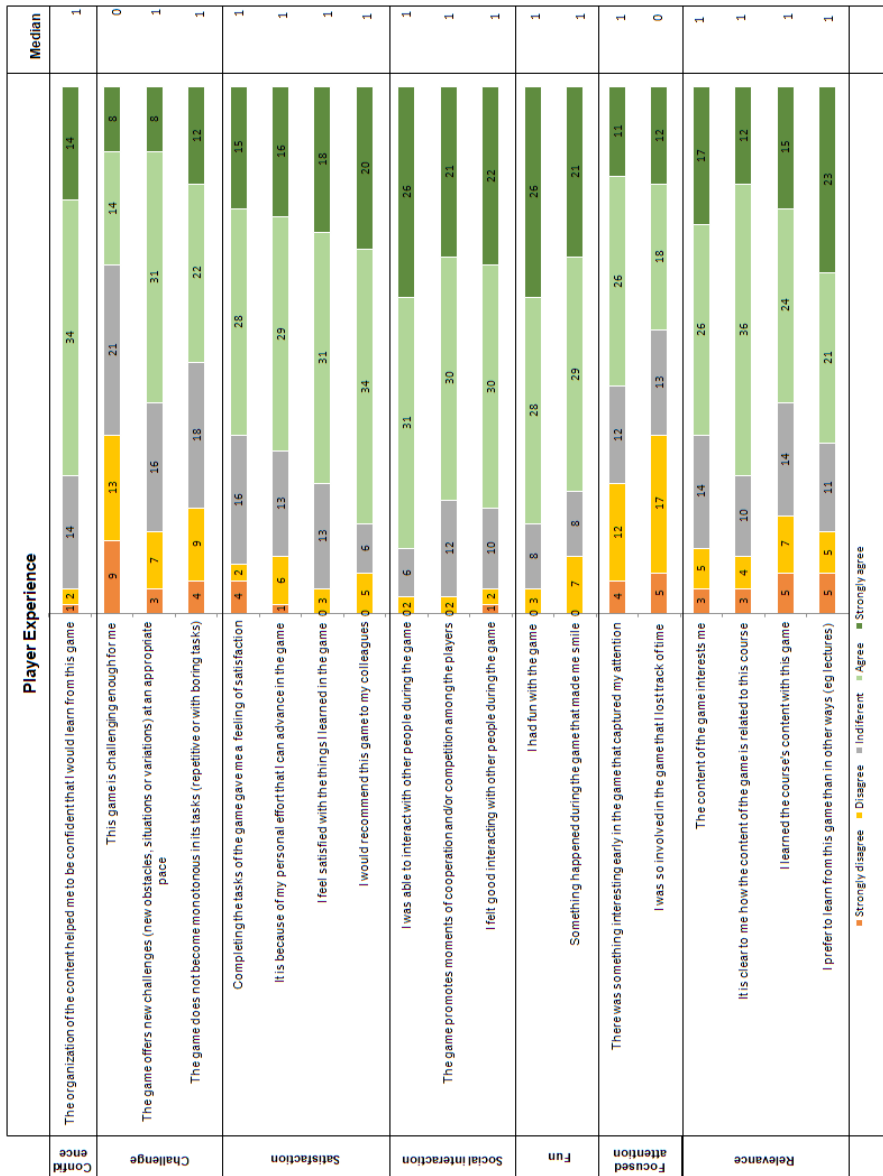


Fig. 7. Frequency and median distribution of items related to the player's experience.

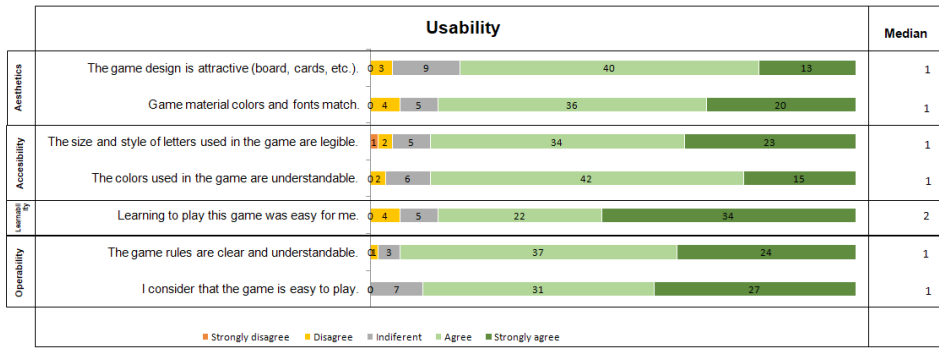


Fig. 8. Frequency and median distribution of items related to usability.

5.2. Is the Game Attractive and Easy to Play?

Regarding usability, the game was evaluated very positively (Fig. 8).

Most students found the game easy to play with clear, easy-to-understand rules. They also agreed that the design of the game material is attractive and accessible especially with regard to colors and typography.

5.3. Does the Game Contribute to the Students Learning?

In order to evaluate the learning, we applied a brief test with 3 questions to assess knowledge on algorithms (Fig. 9).

Two questions are related to content directly addressed by the game (Question 1 and Question 3). Question 1 has been answered correctly by the majority of the students (82%) as well as Question 3 (65%) (Fig. 10).

Question 2 that refers to content more indirectly addressed by the game has been answered correctly only by a small percentage of the students (34%).

These results, while being based on a small number of questions and a without comparative pre-test, can provide a first indication that the game helps to learn concepts related to algorithms.

These learning outcomes are also reflected by the students' comments about their perception (Table 2). Several students reported that they learned about algorithms as well as the need to think and plan the steps as a way to play the game in order to win. Some have also recognized an understanding of how commands and, therefore, programs work. Very few students (6) indicated that they did not learn anything by playing the game.

Select the correct answer. Each question has a single correct answer.
<p>1. An algorithm:</p> <p><input type="checkbox"/> is the internal part of a computer</p> <p><input type="checkbox"/> are the steps necessary to execute a task</p> <p><input type="checkbox"/> are the images that appear on the monitor</p> <p><input type="checkbox"/> is a computer game</p>
<p>2. A computer program:</p> <p><input type="checkbox"/> are the devices (keyboard, mouse, etc.) you use to interact with the computer</p> <p><input type="checkbox"/> are the internal parts of the computer</p> <p><input type="checkbox"/> is the translation of an algorithm into instructions that the computer understands</p> <p><input type="checkbox"/> is the set of commands to play a game on the computer</p>
<p>3. Consider the following steps of an algorithm for brushing teeth:</p> <ol style="list-style-type: none"> 1. Rinse with water 2. Clean the toothbrush 3. Pick up the toothbrush 4. Brush your teeth 5. Store the toothbrush 6. Place toothpaste on the brush <p>What is the correct order of the steps of the algorithm to brush the teeth? (Example answer: 1-2-3-4-5-6).</p> <p>Answer: _ - _ - _ - _ - _ - _</p>

Fig. 9. Questions on the understanding of algorithms.

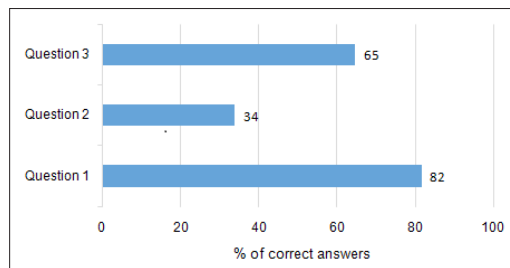


Fig. 10. Percentage of correct answers for each question (total of 65 responses).

Table 2

Answers to the question “What did you learn by playing this game?” with citation frequencies

The way to play the game (11)
Think about what to do later (6)
Algorithms (6)
How commands work (6)
Nothing (6)
Programming (3)
Several things (3)
I learned to pay attention to the movements of colleagues (2)
Board games can be fun (2)
To not just walk forward (2)
Everything (2)
I do not know (2)
In order to win, it is necessary to collaborate with others (1)
Not informed (13)

Table 3
Qualitative comments with citation frequencies

What did you like in the game?	Cute (Kawai) Characters (17) The design (game board, colors etc.) (8) Everything (8) The way the game is played (5) How easy the game is to play (2)? Yes (2) The title of the game and the idea of the game (1) To win the game (1) I had a lot of fun (1) Not informed (24)
What did you think was bad?	Nothing (38) To receive many repeated cards (1) Poor material (1) The game was against me (1) Cards that let you get stuck in the beginning (1) Losing, because I did not receive good cards (1) Not very animating (1) The game could be longer (1) Not being able to pass other players (1) The board game is small; it could have more squares (1) Tokens are not magnetic (1) Not informed (18)
Any further comments?	No (36) I liked that the parts were made from recycling materials (2) I liked the game a lot (1) Top and cute (1) Top Kawaii (1) Poor material (1) Not informed (23)

5.4. General Comments

As the main strengths of the game, students cited the aesthetics of the characters and the attractive design of the game (Table 3). Several students explicitly mentioned the use of Kawaii elements, confirming a design decision in accordance with their preferences with respect to manga books.

Most students did not indicate any need for improvement. Only one student considered the game material poor. However, several students expressed some frustration at having lost the game due to the bad luck they had in relation to the cards they received randomly limiting their ability to select commands.

6. Discussion

The results indicate a positive evaluation of the educational game. In particular, students enjoyed social interaction and had fun while learning with this complementary instruc-

tional method. We also noted that the students participated actively, enjoying the game as a pleasant learning experience. The game was perceived more as a playful activity and less as an instructional activity. While waiting for other groups to finish, several groups even started a second game session rather than doing other activities, such as checking their social networks. These results are in accordance with our goal to teach basic concepts on algorithms in a fun and engaging way. This strength of educational games has also identified by the few evaluations reported of this type of game (Tsarva *et al.*, 2018; Casarotto *et al.*, 2018; Singh *et al.*, 2007).

Students felt confident during the game and perceived a satisfactory effect. Although most students indicated that they felt their achievements were due to learning, some also commented that the odds of winning the game are strongly influenced by the movement cards they receive randomly. Perhaps grouping and distributing the cards per type can minimize this negative impact, preventing a player to receive only cards of the same type of movement (e.g., 5 jump obstacle cards). Similar to Tsarva *et al.* (2018), we observed that the students felt competent and immersed during game play and perceived positive affect.

The students evaluated the usability of the game very positively. Especially the design was cited several times as the main strength of the game. Several students also mentioned the adoption of Kawaii elements. Based on these comments, strongly favoring the cute characters, we also inferred that the game's narrative, helping the animals to cross the forest, was well accepted. This narrative focusing on a positive goal to help was chosen intentionally, rather than a destructive/combat narrative, such as, for example, used in the Flexicard game. The positive assessment of this type of narrative and the use of Kawaii characters has been observed even with a majority of boys participating. Thus, even being a narrative and design typically rather preferred by girls as a strategy to interest them in computing, it is also well accepted by boys. Different to other games that also adopt ludic game tokens such as wooden crab badges (Tsarva *et al.*, 2018), the game material of SplashCode can be created using recycling materials, keeping the costs low. Different also to the majority of the games encountered being sold commercially, the complete game material of SplashCode is available online for free, in order to enable its widespread application in schools with few resources.

In order to facilitate the understanding of the game rules, we kept them as simple as possible. But, still, during the first application, we realized that the initial explanation of how to play the game using slides in a lecture style partially discouraged the students. Therefore, we created a video presenting the game in a playful way. In the following applications using the video, we observed that this not only reduced the time required for the explanation, but also left the students much more excited to play.

Very few students showed any dissatisfaction and/or reported the perception of negative emotions. Only one student considered the material poor, without further explanation.

And while limiting ourselves to a few test questions at the end of the game, we have identified a first indication that the game can help to reinforce the understanding of basic concepts on algorithms in the context of introducing computational thinking.

The students reported that they learned about algorithms recognizing the importance of the sequence of movements in order to win the game. In general, the evaluation of the game SplashCode has provided promising results in terms of player experience, usability, and learning, motivating a broader application of this game in computational thinking education activities in middle school. Being proposed as a quick complementary activity, it can be used in order to reinforce this content in a fun and engaging way.

Threats to validity. The design of the case study applied in this research may cause several threats to the validity of the results. One issue is the lack of a benchmark pre-testing the knowledge of the students before playing the game in order to allow to compare the results with the post-test. This is further exacerbated by the fact that there is no control group to compare the identified effects. However, in this study, we chose to perform a case study rather than an experiment in order to limit interruptions of the normal flow in class. In addition, the fact that the game has been applied with a small set of participants within the same school reduces the possibility of generalizing the results. Yet, considering the exploratory nature of our research, we consider the scientific rigor of a carefully defined case study acceptable. As part of future research we are planning to extend the evaluation of the game in other schools with a larger sample. Another possible threat is that aspects such as fun and satisfaction are difficult to measure and captured by subjective measures. To mitigate this risk, questionnaire items were systematically derived based on the MEEGA+ standardized measurement instrument that was evaluated in terms of validity and reliability on a large scale (Petri *et al.*, 2017).

7. Conclusion

This article presents the development, application and evaluation of a low-cost board game aimed at reinforcing the understanding of basic concepts of algorithms as part of teaching computational thinking in middle school. The game is designed to be adopted quickly (about 15 minutes) in an instructional unit. Results of an application with a total of 65 students in a public school in Florianópolis/Brazil provide a first indication that the game can contribute to the learning of the understanding of algorithms. Students also assessed player experience and usability in a very positive way, showing that the game can be used for a fun, efficient and effective learning experience. Based on these positive results, we intend to broaden the adoption of the game. We also plan to continue the evaluation of the game on a larger scale in a broader variety of contexts in order to improve the generalizability of our results as well as by comparing effects with other instructional methods adopting experimental research designs.

Acknowledgments

The authors would like to thank all students who participated in the application and members of GQS/INCoD/INE/UFSC for their feedback, especially the junior team of the initiative Computing in School. Thanks also to Maria Eduarda for the idea of the name of the game. We would also like to thank the company Involves Technology and Innovation, sponsor of the Young Programming Tutors project, enabling the application and evaluation of the game.

This work was supported by CNPq, a Brazilian government entity focused on scientific and technological development.

References

- All, A., Castellar, E. P. N., Looy, J. V. (2016). Assessing the effectiveness of digital game-based learning: Best practices. *Computers & Education*, vol. 92–93, 90–103.
- Anderson, L. W., Krathwohl, D. R., & Bloom, B. S. (2001). *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*. New York: Longman.
- Astrachan, O., Briggs, A. (2012). The CS Principles Project. *ACM Inroads*, 3(2), 38–42.
- Battistella, P. & Gresse von Wangenheim, C. (2016a). ENgAGED: Um Processo de Desenvolvimento de Jogos para Ensinar Computação. In: *Proc. of the 27th Brazilian Symposium on Informatics in Education, Uberlândia, Brazil* (in Portuguese).
- Battistella, P. & Gresse von Wangenheim, C. (2016b). Games for teaching computing in higher education – a systematic review. *IEEE Technology and Engineering Education Journal*, 9(1), 8–30.
- Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*. New York: Springer New York Dordrecht Heidelberg London.
- Calderón, A. & Ruiz M. (2015). A systematic literature review on serious games evaluation: An application to software project management. *Computers & Education*, 87, 396–422.
- Casarotto, R. I., Bernardi, G., Cordenonsi, A. Z., Duarte Medina, R. (2018). Logirunner: um Jogo de Tabuleiro como Ferramenta para o Auxílio do Ensino e Aprendizagem de Algoritmos e Lógica de Programação. *RENOTE – Revista Novas Tecnologias na Educação*, 16(1) (in Portuguese).
- Connolly, T. M.; Boyle, E. A.; MacArthur, E.; Hainey, T.; Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education*, 59(2), 661–686.
- CSTA. *K–12 Computer Science Standards*, 2017. <<http://k12cs.org>>.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319–340.
- Fu, F., Su, R., & Yu, S. (2009). EGameFlow: A scale to measure learners' enjoyment of e-learning games. *Computers & Education*, 52(1), 101–112.
- Garris, R., Ahlers, R., Driskell, J. E. (2002). Games, Motivation, and Learning: A Research and Practice Model. *Simulation & Gaming*, 33(4), 441–467.
- Gresse von Wangenheim, C. & Shull, F. (2009). To Game or Not to Game? *IEEE Software*, 26(2), 92–94.
- Gresse von Wangenheim, Petri, G., C., & Borgatto, A. F. (2018). *MEEGA+KIDS: A Model for the Evaluation of Educational Games for Computing Education in Secondary School*. Technical Report INCoD/GQS.06.2018.E, INCoD/INE/UFSC, Florianópolis/Brazil.
- Grover, S. & Pea, R. (2013). Computational Thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.
- Guzdial, M. (2008). Education: Paving the way for computational thinking. *Communications of the ACM*, 51(8), 25–27.
- Haddaway, N. R. *et al.* (2015). The role of Google Scholar in evidence reviews and its applicability to grey literature searching. *PloS one*, 10(9).
- International Standard Organization (ISO). (2014). ISO/IEC 25010: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, International Standard.

- Kalelioglu, F., Gülbahar, Y., and Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing*, 4(3), 583–596.
- Kazimoglu, C., Kiernan, M., Bacon, L., Mackinnon, L. (2012). A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming. *Procedia – Social and Behavioral Sciences*, 47, 1991–1999.
- Keller, J. (1987). Development and Use of the ARCS Model of motivational Design. *Journal of Instructional Development*, 10(3), 2–10.
- Kish, L. (1994). Multipopulation survey designs: five types with seven shared aspects. *International Statistical Review*, 62(2), 167–186.
- Mohamed, H. & Jaafar, A. (2010). Development and Potential Analysis of Heuristic Evaluation for Educational Computer Game (PHEG). In: *Proc. of the 5th Int. Conference on Computer Sciences and Convergence Information Technology*, Seoul, South Korea.
- O'Brien, H. L. & Toms, E. G. (2010). The Development and Evaluation of a Survey to Measure User Engagement. *Journal of the American Society for Information Science and Technology*, 61(1), 50–69.
- Petri, G., Gresse von Wangenheim, C., Borgatto A.F. (2018). MEEGA+, Systematic Model to Evaluate Educational Games. In: Lee N. (Eds.) *Encyclopedia of Computer Graphics and Games*, Springer.
- Pfahl, D., Ruhe, G., Koval, N. (2001). An Experiment for Evaluating the Effectiveness of Using a System Dynamics Simulation Model in Software Project Management Education. In: *Proc. of the Int. Symposium on Software Metrics*, London, UK.
- Savi, R., Gresse von Wangenheim, C., Borgatto, A. F. (2011). A Model for the Evaluation of Educational Games for Teaching Software Engineering. In: *Proc. of the Brazilian Symposium on Software Engineering*, São Paulo, Brazil (in Portuguese).
- Sindre, G. & Moody, D. (2003). Evaluating the Effectiveness of Learning Interventions: an Information Systems Case Study. In: *Proc. of the 11th European Conf. on Information Systems*, Paper 80. Naples, Italy.
- Singh, J., Kumar Dorairaj, S., Woods, P. (2007). Learning Computer Programming Using A Board Game – Case Study on C-Jump. In: *Proc. of the Int. Symposium on Information and Communications Technologies*, Kuala Lumpur, Malaysia.
- Sweetser, P. & Wyeth, P. (2005). GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment*, 3(3), 1–24.
- Tsarava, K., Moeller, K., Ninaus, M. (2018). Training Computational Thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games*, 5(2), 25–44.
- Tullis, T. & Albert, W. (2008). *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Burlington: Morgan Kaufmann.
- Unnikrishnan, R., Amrita, N., Muir, A., Rao, B. (2016). Of Elephants and Nested Loops: How to Introduce Computing to Youth in Rural India. In: *Proceedings of the 15th International Conference on Interaction Design and Children*, Manchester, UK.
- Wang, P. S. (2015). *From Computing to Computational Thinking* (1st ed.), New York: Chapman and Hall/CRC.
- Wiebe, E. N., Lamb, A., Hardy, M., & Sharek, D. (2014). Measuring engagement in video game-based environments: Investigation of the User Engagement Scale. *Computers in Human Behavior*, 32, 123–132.
- Wing, J. M., Computational Thinking, *Theoretical Computer Science*, 49(3), 2006.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in Software Engineering*. New York: Springer-Verlag Berlin Heidelberg.
- Yin, R. K. (2017). *Case study research: design and methods* (6th ed.). Thousand Oaks: Sage Publications, Inc.

C. Gresse von Wangenheim is a professor at the Department of Informatics and Statistics (INE) of the Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, where she coordinates the Software Quality Group (GQS) focusing on scientific research, development and transfer of software engineering models, methods and tools and software engineering education. She also coordinates the initiative Computing at Schools that aims at bringing computing education to schools in Brazil. She received the Dipl.-Inform. and Dr. rer. nat. degrees in Computer Science from the Technical University of Kaiserslautern (Germany), and the Dr. Eng. degree in Production Engineering from the Federal University of Santa Catarina. She is also PMP – Project Management Professional and MPS.BR Assessor and Implementor.

G. Araújo e Silva de Medeiros, is a master student of the Graduate Program in Education (PPGE) at the Federal University of Santa Catarina (UFSC), a research student at the initiative Computing at Schools/INCoD/INE/UFSC and an educational technology teacher in Florianópolis, Brazil.

R. Missfeldt Filho, is an undergraduate student of the Computer Science course at the Federal University of Santa Catarina (UFSC) and a scholarship student at the initiative Computing at Schools/INCoD/INE/UFSC.






G. Petri is a professor at the Polytechnic School of the Federal University of Santa Maria (UFSM), Brazil. He holds a PhD. in Computer Science at the Federal University of Santa Catarina (UFSC) and a master's degree in Computer Science at the Federal University of Santa Maria. His main research interests are in the area of computing education and educational games.








F. da Cruz Pinheiro, is a master student of the Graduate Program in Computer Science (PPGCC) at the Federal University of Santa Catarina (UFSC) and a research student at the initiative Computing at Schools/INCoD/INE/UFSC.




M.N.F. Ferreira, is a master student of the Graduate Program in Design (PPGDEG) at the Federal University of Santa Catarina (UFSC) and a research student at the initiative Computing at Schools/INCoD/INE/UFSC.

J.C.R. Hauck holds a PhD in Knowledge Engineering and a Master's Degree in Computer Science from the Federal University of Santa Catarina (UFSC) and a degree in Computer Science from the University of Vale do Itajaí (UNIVALI). He held several specialization courses in Software Engineering at Unisul, Univali, Uniplac, Uniasselvi, Sociesc and Uniarp. He was a visiting researcher at the Regulated Software Research Center – Dundalk Institute of Technology – Ireland. He is currently a Professor in the Department of Informatics and Statistics at the Federal University of Santa Catarina.

Appendix A. Overview on board/card games for teaching computational thinking in K-12

Name	Image	Short description	Age	Number of players	Duration	Language	Approx. cost
Bits & Bytes (www.bitsandbytes.cards)		The goal of the game is for each player to guide their character (program) to their home planet, Ram, by issuing instructions (turn right, turn left, move forward, turn around). At the same time, they have to avoid walls, bugs and the dreaded CPU.	4-9	2-4	--	English	US\$20
C-Jump (c-jump.com) (Singh et al., 2007)		Skiers and snowboarders line up at the start location and race along the ski trails. Spaces on the board show statements of programming language. First player to move all skiers past the finish line is the winner. Players calculate number of steps in the move, including addition, subtraction, division, and multiplication of small numbers. The game helps to develop understanding of a complete computer program, formed by logical sequences of commands.	11+	2-4	30 min	English	US\$35
Coder Bunnyz (http://www.coderbunnyz.com)		Players program their robot bunny game token with their code cards to eat the carrot and reach the destination before the others do.	4+	1-4	10-20 min	English	US\$35
Code Island Monkey (http://codemonkeyplanet.com)		As the wise leader of a tribe of monkeys, players have to guide three monkeys safely around the board into the banana grove. Therefore, players have to use concepts like conditional statements, looping, booleans, assignment operators etc. in order to move their monkeys, dodge quicksand traps, and score fruits along the way.	8+	2-4	45 min	English	US\$35
Code & Go Mouse Mania (https://www.learningresources.com/product/code-go-mouse-mania-board-game-2863.do)		Players draw coding cards and compute their mice towards cheese wedges scattered across the board navigating their mice away from barriers and toward warp tunnels	5+	2-4	--	English	US\$20

Name	Image	Short description	Age	Number of players	Duration	Language	Approx. cost
Codemaster – programming logic game (https://www.thinkfun.com/products/code-master)		Players have to use programming logic to help their avatar collect the crystals and land at the portal traveling to an exotic world. For all 60 levels, players will have to use a specific sequence of actions will lead them to success.	8+	1	--	English	US\$20
Coding Farmers (codingfarmers.net)		Players use action cards like “move forward two spaces” expressed in regular English, and Java code, rolling a dice and maneuvering around obstacles.	7+	2-4	45-60 min	English	US\$15
Crabs & Turtles – Treasure hunt (Tsarava <i>et al.</i> , 2018)		Players have to manipulate colored game pieces representing turtles and crabs to figure out the most efficient way to collect treasures placed on the grid squares of the game board. To move a crab or turtle, players need to create effective sequences of commands on a sequence board, which represent specific coding concepts.	6+	teams of two	--	English, German, Greek	--
Crabs & Turtles – The Race (Tsarava <i>et al.</i> , 2018)		Players have to reach the end of the game board by solving and handling math related riddles and events. To do so, players have to manipulate constants and changing values of variables or make decisions based on conditionals.	6+	--	--	English, German, Greek	--
FlexiCard (http://www.computacionaL.com.br/index.php#FlexiCard)		Players move on a maze to solve a challenge, such as reaching for the star, catching a key and open the door, or knock down zombies and walk out the door, etc. using command cards, which are executed by a master player.	4+	2-4	15 min	Brazilian Portuguese	Gratuito (CC BY-NS-AS 4.0)
Logrunner (Casarotto <i>et al.</i> , 2018)		The game represents a race in which players pass movement instructions to their characters (runners) in order to complete objectives (checkpoints) in a world represented by a checkered board.	6+	2-4	35-65 min	Brazilian Portuguese	--
Potato Pirates (https://www.potatopirates.game)		Players need to acquire all potato king cards or eliminate all other players. Therefore, they can roast, mash, or fry your opponents and sink their boats by using programming concepts such as functions, loops, and conditionals to fortify their attacks.	6+	3-6	20-40 min	English	US\$30

Name	Image	Short description	Age	Number of players	Duration	Language	Approx. cost
Robot Turtles http://www.thinkfun.com/robot-turtles		<p>Players dictate the movements of their robot turtle tokens on a game board by playing code cards (forward, left and right). When a player's robot turtle reaches a jewel, they win. If they make a mistake, they can use a bug card to undo a move. The game has many levels so, as the players advance, they will encounter obstacles like ice walls and use more complex code cards.</p>	4+	2-5	15 min	English	US\$25
Robo Rally (https://en.wikipedia.org/wiki/RoboRally)		<p>Players control their robot token with game cards, which reveal directions on how the robots can move through the Robo Rally automobile factory. Players face obstacles like industrial lasers, gaping pits, and moving conveyor belts. In order to win the game, players need to be the first to reach each of the checkpoints in numerical order.</p>	12+	2-8	45-120 min	English, French, German, Dutch	US\$30
Robot Wars		<p>Players get to shoot other players' robots in a war scenario. Commands for bots are written in Java, but readable as simple English. The first player to reach the middle grid of the boardwins.</p>	7+	2-4	40-60 min	English	US\$20